PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-344757
(43)Date of publication of application : 29.11.2002

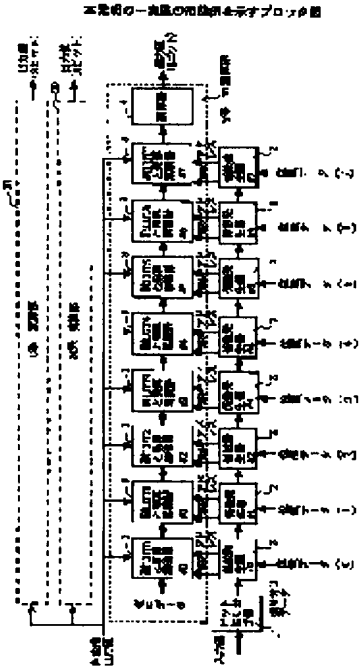
(51)Int.Cl. H04N 1/60
B41J 2/525
G06T 1/00
G06T 1/20
H04N 1/46

(21)Application number : 2001-146327
(22)Date of filing : 16.05.2001

(71)Applicant : KONICA CORP
(72)Inventor : KO HIROTETSU

(54) COLOR INTERPOLATION METHOD AND COLOR INTERPOLATION APPARATUS

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a color interpolation method, with which a high-speed arithmetic operation can be obtained and a color interpolation apparatus.
SOLUTION: An entire color conversion table is divided into a plurality of color conversion tables, and the tables are used in series to conduct color interpolation. Arithmetic circuits are arranged in series, so as to utilize the same unit and accelerate high-speed processing by pipeline.



(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－344757
（P2002－344757A）
(43)公開日　平成14年11月29日(2002.11.29)

(51)Int.Cl.⁷		識別記号	FI			テーマコード（参考）
H04N	1/60		C06T	1/00	510	2C262
B41J	2/525			1/20	C	5B057
G06T	1/00	510	H04N	1/40	D	5C077
	1/20			1/46	Z	5C079
H04N	1/46		B41J	3/00	B	

審査請求　未請求　請求項の数6　ＯＬ　（全 12 頁）

(21)出願番号　特願2001－146327(P2001－146327)

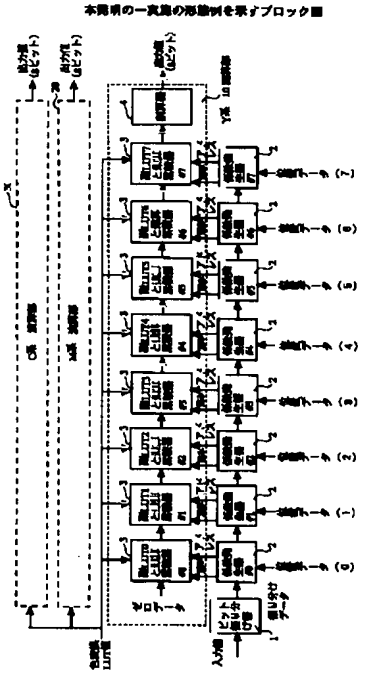
(22)出願日　平成13年５月16日(2001.5.16)

(71)出願人　000001270
コニカ株式会社
東京都新宿区西新宿１丁目26番２号
(72)発明者　洪　博哲
東京都八王子市石川町2970番地　コニカ株式会社内
(74)代理人　100085187
弁理士　井島　藤治　（外１名）

最終頁に続く

(54)【発明の名称】　色補間方法及び色補間装置

(57)【要約】　（修正有）
【課題】　色補間方法及び色補間装置に関し、高速演算が可能な色補間方法及び色補間装置を提供する。
【解決手段】　全体の色変換テーブルを複数の色変換テーブルに分割して持ち、それらをシリアルに用いて色補間を行なうように構成する。演算回路をシリアルにならべることにより、同じユニットが利用でき、パイプライン化による高速化ができる。



【特許請求の範囲】
【請求項１】　全体の色変換テーブルを複数の色変換テーブルに分割して持ち、それらをシリアルに用いて色補間を行なうことを特徴とする色補間方法。
【請求項２】　前記色変換テーブルは、ｎ色入力の時に２のｎ乗個に分割することを特徴とする請求項１記載の色補間方法。
【請求項３】　前記分割された色変換テーブルが同時に使われない組み合わせについては、パラレルに並べることを特徴とする請求項２記載の色補間方法。
【請求項４】　全体の色変換テーブルを複数の色変換テーブルに分割して持ち、２のｎ乗より少ない数で補間する場合、当該色変換テーブルを使用しない部分に０の重みを加えることを特徴とする請求項１記載の色補間方法。
【請求項５】　前記補間演算点数を可変できることを特徴とする請求項１又は３の何れかに記載の色補間方法。
【請求項６】　全体の色変換テーブルを複数の色変換テーブルに分割して持ち、それらをシリアルに用いて計算する部分を有することを特徴とする色補間装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は色補間方法及び色補間装置に関し、更に詳しくは高速化した色補間方法及び色補間装置に関する。色変換方法として、多次元のＬＵＴと補間演算器による演算が知られている。米国特許第４８３７７２２号では、高速演算のために補間演算器をパラレルに並べて計算する方法が提案されている。
【０００２】
【従来の技術】カラープリンタ、カラースキャナ、カラー複写機等では、光電走査によって得た色分解ディジタル画像データの色修正を行なうことが多い。上記のようなディジタル画像データの色修正は演算により行なわれる場合もあるが、予め色分解画像データの組み合わせ（例えば赤Ｒ，緑Ｇ，青Ｂの３原色画像データの組み合わせ）に対応する修正済み画像データの組み合わせ（例えばイエローＹ，マゼンタＭ，シアンＣ，ブラックＫのインキ量）を変換テーブルとして記憶させておき、修正前のデータの組み合わせをアドレス指定信号として、修正済みのデータの組み合わせ（変換出力データ）を読み出すように構成される場合もある。
【０００３】前述のような変換テーブルを用いたデータ変換では、入力データの組み合わせ全てに対応する修正データの組み合わせを記憶するようにすると、膨大な記憶容量が必要となって、変換テーブルとして用いるメモリのコストアップになるので、変換出力データを適宜間引いて記憶させ、間引いた部分に相当する入力データの組み合わせに対応する修正データについては補間演算により求めることが一般に行われている。
【０００４】上記のように、変換テーブルを用いたデータ変換を、補間演算を伴って行わせる技術としては、従来、特公昭５８－１６１８０号公報及び特開昭６３－１６２２４８号公報等に開示されるようなものがある。特公昭５８－１６１８０号公報では、３次元の入力ディジタルデータが構成する３次元の空間を複数の単位立方体で分割し、該複数の立方体の頂点それぞれに対応させて変換出力データを記憶させて変換テーブルを構成する。
【０００５】そして、前記単位立方体を、該単位立方体の８頂点の中の４頂点からなる複数（５又は６個）の４面体に分割し、３次元の入力データに対応する点を含む４面体の４頂点それぞれにおける変換出力データを読み出し、該４個の変換出力データを補間演算することで、３次元入力データに対する最終的な変換出力データを出力するようにしている。
【０００６】また、特開昭６３－１６２２４８号公報には、前記立方体をそのまま用い、３次元の入力データに対応する点を含む立方体の８頂点に対応して読み出される８個の変換出力データから補間演算により最終的な変換出力データを出力するようにしている。上記の特公昭５８－１６１８０号公報及び特開昭６３－１６２２４８号公報等に開示されるデータ変換では、補間演算を用いることで変換テーブルに記憶させる変換出力データの数を少なくしてメモリ容量を小さく抑えると共に、非線形の変換も小さいな誤差で行なえるものであり、またハードウェアも比較的小規模ですみ、比較的高速な回路が実現できる。
【０００７】
【発明が解決しようとする課題】パラレルに演算を行なう場合、全体として設計しなくてはならないため、回路が複雑になるという問題がある。また、一つのユニットを繰り返し利用できず、例えば乗算器と足し算器を別々に用いることになる。また、補間演算点数はハードウェアで固定的である。しかしながら、例えば入力データがＬ*ａ*ｂ*では８点補間が最適であり、入力データがＲ，Ｇ，ＢやＹ，Ｍ，Ｃの場合には４点補間が最適であることが知られている。この場合、それぞれに最適な補間演算が行われないことになる。
【０００８】本発明はこのような課題に鑑みてなされたものであって、高速演算が可能な色補間方法及び色補間装置を提供することを目的としている。即ち、第１にメモリの総量を最小限にして、かつ最大の計算能力を実現し、第２に同じ形式のユニットを並べることで、部品及びタイミング設計を容易にし、第３に容易に補間に必要な格子点数を変更できる仕組みを提案する。
【０００９】
【課題を解決するための手段】（１）請求項１記載の発明は、全体の色変換テーブルを複数の色変換テーブルに分割して持ち、それらをシリアルに用いて色補間を行なうことを特徴とする。
【００１０】このように構成すれば、演算回路をシリア

ルに並べることで、同じユニットが利用でき、かつパイプライン化できるので高速化することができる。

（２）請求項２記載の発明は、前記色変換テーブルは、ｎ色入力の時に２のｎ乗個に分割することを特徴とする。

【００１１】このように構成すれば、補間演算の点数を自由に設定することができる。

（３）請求項３記載の発明は、前記分割された色変換テーブルが同時に使われない組み合わせについては、パラレルに並べることを特徴とする。

【００１２】このように構成すれば、遅延を少なくすることができる。

（４）請求項４記載の発明は、全体の色変換テーブルを複数の色変換テーブルに分割して持ち、２のｎ乗より少ない数で補間する場合、当該色変換テーブルを使用しない部分に０の重みを加えることを特徴とする。

【００１３】このように構成すれば、簡単な構成でバイパスするのと同等の効果が得られる。

（５）請求項５記載の発明は、前記補間演算点数を可変できることを特徴とする。

【００１４】このように構成すれば、入力データに応じて最適な補間演算手段が選択できる。

（６）請求項６記載の発明は、全体の色変換テーブルを複数の色変換テーブルに分割して持ち、それらをシリアルに用いて計算する部分を有することを特徴とする。

【００１５】このように構成すれば、演算回路をシリアルに並べることで、同じユニットが利用でき、かつパイプライン化できるので高速化することができる。

【００１６】

【発明の実施の形態】以下に、図面を参照して本発明の実施の形態例を詳細に説明する。図１は本発明の一実施の形態例を示すブロック図で、３次元入力の場合を例にとっている。図において、１は入力データ２４ビットを受けて、ビットを振り分けるビット振り分け器である。入力２４ビットデータは、例えばRが８ビット、Gが８ビット、Bが８ビットである。２はビット振り分け器１からの出力信号を受けて重みとアドレスを出力する係数発生器である。該係数発生器２は、＃０～＃７までの８個存在する。そして、これら係数発生器２は＃０が＃１に入り、＃１が＃２に入るという具合に直列に接続されている。

【００１７】１０は副ＬＵＴと乗算累積器を８個用いた演算部である。３が副ＬＵＴと乗算累積器（以下単に演算器と略す）であり、図に示すように＃０～＃７まで８個直列に接続されている。＃０の演算器には入力としてゼロデータが入力されている。該＃０の演算器の出力は＃１の演算器に入り、＃１の演算器の出力は＃２の演算器に入る。以下、同様に接続され、演算出力が累積されていくようになっている。演算部１０において、４は最終段の演算器の出力を受けて重みの総和で割算して出力（８ビット）を得る割算器である。

【００１８】図に示す演算部１０は、Ｙ，Ｍ，Ｃ毎に設けられている。即ち、１０はＹ系の演算部であり、２０はＭ系の演算部であり、３０はＣ系の演算部である。そして、それぞれの演算部からは、８ビットの演算出力が得られるようになっている。Ｋや特色出力があれば、このユニットが増える。色変換ＬＵＴ値は、それぞれの演算部に共通に入力されている。この色変換ＬＵＴ値は、それぞれの演算器３毎に分割されて与えられている。また、各係数発生器２からは、対応する演算器３に重みとアドレスが与えられている。図に示す回路は、図示しないタイミングクロックにより同期して動作するようになっている。このように構成された回路の動作を説明すれば、以下の通りである。

【００１９】図１に示す実施の形態例は、三角錐を用いた４点補間法を実現する回路である。図２は三角錐を用いた４点補間の説明図である。立方体を三角錐で分割すると、６個の三角錐に分割される。そして、それぞれの条件毎に補間式を求める。例えば$x \geq y \geq z$の場合、当該三角錐内の任意の点Ｐの補間値p_{xyz}は次式で表される。

【００２０】

$$p_{xyz}=p_a(1-x)+p_b(x-y)+p_c(y-z)+p_g z \quad (1)$$

但し、この式では格子間隔が１の場合を示している。ここで、ｐａ～ｐｇは三角錐の各頂点の値を示す。図１に示す回路は、ハードウェア的に（１）式を実現するものである。

【００２１】本来の３次元補間用ＬＵＴをＮ×Ｎ×Ｎ（Ｎは偶数）としたとき、８つに分割して、それぞれのブロックに転送する。この分割方法は、図３に示すように、一つおきに並べたデータを８つの（Ｎ／２）×（Ｎ／２）×（Ｎ／２）のＬＵＴに分割するものである。図３において、（ａ）に示すＬＵＴ全体を○、□、△、×の点を一つおきにとっていき、各々の点で構成されるデータで分割すると、図３の（ｂ）～（ｅ）に示すように分割される。図は２次元の場合を示す。３次元の場合には８個に分割される。

【００２２】このように分割されたＬＵＴを副ＬＵＴｎと呼ぶことにする。ここで、ｎは０から７までの数字である。この副ＬＵＴは、ｎを０から（（Ｎ／２）－１）とし、その２進数をｉ，ｊ，ｋとしたとき、

$$LUTn[I][J][K]=LUT[2\cdot I+i][2\cdot J+j][2\cdot K+k] \quad (2)$$

により設定されるサイズ１／８のＬＵＴである。ここで、Ｉ，Ｊ，Ｋは０～７のＲ，Ｇ，Ｂのアドレス、ｉ，ｊ，ｋは０か１の値である。（２）式は、副ＬＵＴが右辺で表される元のＬＵＴの（２）式に示すアドレスと対

応していることを示す。つまり、左辺の副ＬＵＴは右辺に示すＬＵＴのアドレスで表される場所からとってくるものであることを示す。このように、本発明によれば、色変換テーブルをｎ色入力の時に２のｎ乗個に分割することで、補間演算の点数を自由に設定することができる。

【００２３】ここで、Ｎ＝１６として説明する。図１において、入力されたＲ，Ｇ，Ｂ値は、ビット振り分け器１で、各色上位値と下位値、及び上下判定値に分けられる。図４はビット振り分け器１の一実施の形態例を示すブロック図である。入力値は、Ｒ，Ｇ，Ｂ毎に８ビットの合計２４ビットである。ＲはＲチャネル４０に入り、ＧはＧチャネル４１に入り、ＢはＢチャネル４２に入る。

【００２４】例えば、Ｒチャネル４０において、５０は８ビットのＲデータを受ける上位／下位ビット振り分け器、５１は該上位／下位ビット振り分け器５０の出力を受ける上半位／下半位振り分け器である。上位／下位ビット振り分け器５０からは３ビットの上位値（０～７）が出力され、上半位／下半位振り分け器５１からは１ビットの上下判別値と５ビットの下位値（０～１６）が出力される。以上の構成は、他のＧチャネル、Ｂチャネルについても同様である。Ｒチャネル、Ｇチャネル、Ｂチャネルから出力される５ビットの下位値は三角錐コード発生器４３に入り、該三角錐コード発生器４３からは、３ビットの三角錐コード（０～５）が出力される。三角錐コードは、図２に示す三角錐に０～５までのコードを振り、各三角錐を識別するものである。三角錐発生コードの組み合わせは、図５に示す通りである。例えば、Ｂ≧Ｒ≧Ｇの場合の三角錐コードは“４”である。

【００２５】ここで、図４において示す計算式で上位値、上下判別値、下位値が計算される。例えば、上位値は入力値／３４で表され、１次下位値は入力値を３４で割った余り（％３４）で表され、上下判別値は入力値／１７で表され、更に、最終の下位値は入力値％１７で表される。そして、３色の最終の下位値を用いて補間演算に使用される三角錐（４点の組み合わせ）の位置を決定するコードを三角錐コード発生器４３から出力する。ここで出力されたデータは、各演算と同期して次の係数発生器２（＃０）に転送される。

【００２６】図６は係数発生器の一実施の形態例を示すブロック図である。係数発生器２は副ＬＵＴｎに対するアドレスと補間演算用重み係数を出力して副ＬＵＴｎと乗算器のブロック（演算器）３に送る。ビット振り分け器１からのＲ，Ｇ，Ｂ毎の上位値（０～７）は加算器６１に入り、下位値（０～１６）と、上下判別ビット（０～１）はアドレス発生／重み発生器６０に入る。アドレス発生／重み発生器６０には、ＬＵＴ　ＩＤ[０～７]も入っている。このＬＵＴ　ＩＤは図１に示す位置データと同じものである。

【００２７】アドレス発生／重み発生器６０からは、Ｒ，Ｇ，Ｂ毎の増分アドレスと重み係数が発生される。ここで、増分アドレスは、ＬＵＴのアドレスを指定する時に、最も小さい値を基底にして必要な軸に＋１するものである。例えば、２次元で示すと（３，４）が基底アドレスだとすると、（４，４）、（４，５）のように、＋０又は＋１を行なう。これにより、三角錐の位置を決めるものである。ここで、三角錐の位置を決めるとは、ＬＵＴの格子のいずれを用いるか（格子のいずれかの点で構成される三角錐を用いるか）を決定することである。

【００２８】Ｒ，Ｇ，Ｂ毎の上位値（０～７）と増分アドレス（０～１）は加算器６１に入り、該加算器６１からはＲアドレス（０～７）が発生される。この構成は、残りのＧチャネル、Ｂチャネルについても同様である。Ｒチャネル、Ｇチャネル、Ｂチャネルからは、それぞれのアドレスが発生して演算器３に入り、また重み係数（０～１７）が発生して演算器３に入る。一方、上位値、下位値、上下判別ビット、三角錐コードは、次の係数発生器２に入る。

【００２９】図７は係数発生器中のアドレス／重み発生器６０の一実施の形態例を示すブロック図である。Ｒ，Ｇ，Ｂ毎の下位値（０～１６）は、差分計算部７０に入る。該差分計算部７０は、Ｒ，Ｇ，Ｂ各下位値と差分計算指示器７１からの４ビット出力を受けて、重み係数（０～１７）を発生する。差分計算指示器７１は、図８に示すようなテーブルを持っている。そして、ＬＵＴコード毎に三角錐コードに対応した重みが割り振られている。

【００３０】ＬＵＴコードは、０００、００１、０１０、０１１、１００、１０１、１１０、１１１に区分され、それぞれの三角錐毎に重みが与えられている。各三角錐毎に、重みは４個与えられている。例えば、三角錐コード“３”の場合は、ＬＵＴコード０００が２、０１０が７、１１０が８、１１１が１２である。そして、各三角錐毎に４個の重みが与えられている。各三角錐毎に４個の重みがあり、残りが０であるのは、（１）式が４個の乗算項からなっているのと対応している。つまり、図１の演算器３は８個存在するが、実際に演算を行なっているのは４個であることを示している。このように、本発明によれば、全体の色変換テーブルを複数の色変換テーブルに分割して持ち、２のｎ乗より少ない数で補間する場合、当該色変換テーブルを使用しない部分に０の重みを加えることで、簡単な構成でバイパスするのと同等の効果が得られる。

【００３１】図７において、７２は排他的論理和回路であり、ＬＵＴＩＤ[０～７]とＲ，Ｇ，Ｂそれぞれの上下判別ビットを受けて３ビットのＬＵＴコードを出力し、差分計算指示器７１に与える。差分計算指示器７１は、３ビットのＬＵＴコードと三角錐コードを受けて、図８

に示す4ビットの重みデータを出力し、差分計算部70に与える。

【0032】該差分計算部70は、R，G，B毎の下位値と、差分計算指示器71からの4ビットの重みデータを受けて、図9に示すような差分計算を行ない、重み係数（0～17）を出力する。アンド（AND）回路73は、R，G，B毎の上下判別ビットと、排他的論理和回路72からの3ビット出力を受けて、R，G，B毎の増分アドレス（0～1）を出力する。

【0033】上述したように、係数発生器2は、図6に示したように、副LUTnに対するアドレスと補間演算用重み係数を出力して、副LUTnと乗算累積器のブロック（演算器）3に送る。アドレス発生／重み発生器60は、ビット操作及び図8、図9に示す処理を行なう。これにより、重み係数及び副LUTnに対する増分アドレスを計算する。

【0034】本発明では、元のLUTを複数の副LUTに分割しているため、例えば上下判定値が0の方向では増分は発生せず、上下判定値が1の場合、選択するLUTの種類が変更され、更に増分アドレスが1となる。各色増分アドレスは、各色上位値と加算される。ただし、上限の制限がついている。この場合、副LUTの最大アドレスの7を超えないようにする（図6参照）。

【0035】副LUTnと乗算累積器（演算器）3では、前段のデータを受け取り、副LUTnに対するアドレスにより出力されたLUTデータと重みを乗算した上、累積して次段に送り出す。この場合、初段の演算器3には、0データが与えられる。当該副LUTnに対して重みが発生しない場合（使用しない場合）は、重み0が与えられる。この場合、0との乗算を行なう代わりにセレクタを用いて当該段はバイパスするようにしてもよい。

【0036】図10は副LUTnと乗算器（演算器）3の一実施の形態例を示すブロック図である。図において、80はアドレス（3×3ビット）て色変換LUT値を受けて入力アドレスに対応するデータを出力する副LUTnである。該副LUTnは8ビットのデータを出力する。81は、係数発生器2からの重み係数（5ビット）と副LUTn80からの出力データを受けて乗算を行なう乗算器である。該乗算器81からは、13ビットのデータが出力される。82は、前段の演算器3からのデータ（8+5ビット）と乗算器81の出力（13ビット）とを加算する加算器である。該加算器82の出力（13ビット）が当該段の演算器3の出力となる。ここで、副LUTnの出力と、乗算器81の出力と、加算器82の出力はラッチ構成となっており、同期クロックでデータが左から右にシフトするようになっており、高速動作が可能となっていれる。このように構成された回路の動作を説明すれば、以下の通りである。

【0037】係数発生器2から重み係数（5ビット）とアドレス（3×3）ビットが演算器3に入力される。この内、アドレスは副LUTnに入って、該副LUTnからは8ビットの変換データが出力される。乗算器81は、この副LUTnの出力と重み係数（5ビット）を乗算する。この乗算結果は、続く加算器82で前段からのデータと加算され、その出力（8+5ビット）は後段の演算器3に入るようになっている。

【0038】この演算処理を8段行なうことで、重み係数の総和と最終的な値との乗算値が算出される。この乗算値は、割算器4（図1参照）で重み係数の総和で割り算されて最終的な出力となる。割算器4は、ロジックで作成しても、LUT化してもよい。16×16×16の格子点で4点補間の場合、重みの総和は17になる。このように、本発明によれば、演算回路をシリアルに並べることで、同じユニットが利用でき、かつパイプライン化できるので高速化することができる。

【0039】上述したような処理を行なう時に、各ブロック、更には各ブロックの内部を数段に分割してパイプライン化することにより、高速なスループットが実現できる。例えば、図10に示すようにラッチを設けてパイプラインの幅を狭くすることにより高速化される。但し、この場合は最初の計算値の出力まで遅延が発生する。各ブロック毎にパイプライン化した場合、少なくとも8クロック、内部を更に数段に分割した場合は、その段数を乗算したクロック数だけ遅延する。何れの場合も、最初にデータが出力されるまでの遅延時間は、段数により変化するが、一旦出力された後は、連続して出力されるので、スループットには差は生じない。

【0040】上述の実施の形態例では、補間演算点数を8個設けた場合を示したが、本発明はこれに限るものではなく、補間演算点数を可変することができる。これにより、入力データに応じて最適な補間演算手段が選択できる。

【0041】図11は本発明の他の実施の形態例を示すブロック図である。図1と同一のものは、同一の符号を付して示す。図において、1は入力データ（R，G，B各8ビットで合計24ビット）を受けてビット振り分けを行なうビット振り分け器、2は入力データを受けて重みとアドレスを発生する係数発生器で、#0～#7まで8個存在する。#0の係数発生器2には位置データ（0）とビット振り分け器1の出力が入り、#1の係数発生器2には、位置データ（1）と#0の係数発生器2の出力が入り、#2の係数発生器2には、位置データ（2）と#0の係数発生器2の出力が入り、#3の係数発生器2には、位置データ（4）と#0の係数発生器2の出力が入り、#4の係数発生器2には、位置データ（3）と#3の係数発生器2の出力が入り、#5の係数発生器2には、位置データ（5）と#3の係数発生器2の出力が入り、#6の係数発生器2には、位置データ（6）と#2の係数発生器2の出力が入り、#7の係数

発生器２には位置データ（７）と＃６の係数発生器２の出力が入っている。

【００４２】１０１は副ＬＵＴと乗算累積器（演算器）で、＃０と＃１の２個存在する。＃０の演算器１０１は副ＬＵＴ０と乗算累積器で構成され、＃１の演算器１０１は副ＬＵＴ７と乗算累積器で構成される。そして、＃０の演算器１０１には＃０の係数発生器２からの重み係数とアドレスが入力され、ゼロデータが入力される。＃１の演算器１０１には、＃７の係数発生器２からの重み係数とアドレスが入力され、＃１の加算器１０３の出力が入力される。

【００４３】１０２は副ＬＵＴと乗算器とで構成される演算器であり、＃０～＃５まで６個存在する。＃０の演算器１０２は副ＬＵＴ１と乗算器とで構成され、＃１の演算器１０２は副ＬＵＴ２と乗算器とで構成され、＃２の演算器１０２は副ＬＵＴ４と乗算器とで構成され、＃３の演算器１０２は副ＬＵＴ３と乗算器とで構成され、＃４の演算器１０２は副ＬＵＴ５と乗算器とで構成され、＃５の演算器１０２は副ＬＵＴ６と乗算器とで構成される。

【００４４】１０３は加算器であり、＃０と＃１の２個存在する。＃０の加算器１０３には＃０の演算器１０１の出力と、＃０の演算器１０２の出力と、＃１の演算器の出力と、＃２の乗算器１０２の出力が入っている。＃１の加算器１０３には、＃０の加算器１０３の出力と、＃３の演算器１０２の出力と、＃４の演算器１０２の出力と、＃５の演算器１０２の出力が入っている。そして、＃１の加算器１０３の出力は、＃１の演算器１０１に入り、該＃１の演算器１０１の出力は割算器４に入っている。そして、該割算器４で、累積値が重み係数の総和で割り算され、８ビットのデータとして出力される。これら演算器と加算器と割算器とで演算部１００を構成している。このような構成の演算部は、色毎にｎ個存在する。

【００４５】図１に示す演算器の並べ方では、最低８個のパイプライン分だけ遅延するが、図１１のような演算器の配列にすると、遅延を半分にすることができる。図１に示す各ブロックの副ＬＵＴｎで、同時に発生しない組み合わせがあるため、それらをパラレルにまとめる。副ＬＵＴ０と副ＬＵＴ７はどの三角錐の場合にも用いられるが、副ＬＵＴ１，２，４及び副ＬＵＴ３，５，６の組み合わせ内では同時に使用されることがない。

【００４６】このようなグループでは、図１１に示すように各ブロック内の累積器を省略して、図のように４入力の加算器１０３を用いることができる。この場合、３種類の入力は全て加算される。この計算は、加算器ではなくオア（ＯＲ）をとっても構わない。値を発生するのは１つだけであるためである。その他には、セレクタを用いることも可能である。

【００４７】この実施の形態例によれば、遅延を少なくすることができる。図１に示す実施の形態例では、８種類の色変換ＬＵＴに対してアクセスするりことができるため、この重みをコントロールすることで、補間方法を容易に変更することができる。これを用いて、第１の実施の形態例の４点補間と８点補間を切り替え可能にすることができる。

【００４８】８点で補間する方法としては、特開平８－１１４８７０号公報（多次元補間方法及び装置）があるが、その手法を用いると重み係数が単純化され、好都合である。但し、重みの総和が４倍になるので、計算のビット幅はこれに合わせて増やす必要がある。更に、最終段での割算器の分母パラメータも可変にする。後述のビットシフトを用いる場合には、ビットシフト量を変更する。入力がＲ，Ｇ，Ｂ又はＹ，Ｍ，Ｃの場合には、４点補間を用いてＬ*ａ*ｂ*では、８点補間を用いるとよい。このようにすることで、入力のグレー付近が正確に補間される。

【００４９】ここで、テーブル出力を、予め（２のｎ乗÷データ格子間隔）倍することで、補間値をビットシフトにより得るようにすることができる。例えば、出力を９ビットにして、最大値を４８０（２５５×３２／１７倍）にすると、５ビットシフトすればよく、大規模になりがちな割算器を省略することができる。但し、ＬＵＴの容量が増える。また、最大値を２４０（２５５×１６／１７倍）にした場合、４ビットシフトになるが、計算精度が若干低下する。

【００５０】この方法をソフトウェアでも実現することができる。一つのＭＰＵでは効果がないが、複数のＭＰＵがある場合、それ毎に各ステップの計算を行なうようにすることで、この構成の色補間演算を効果的に行なうことができる。例えば、ネットワーク上に接続されたコンピュータでリレーしながら計算することもできる。４次元の場合には、１６個の演算器を並べることで、同様の処理が可能で、Ｎ次元の場合には、２のＮ乗のユニットを並べることで実現することができる。

【００５１】

【発明の効果】以上、詳細に説明したように、本発明によれば、以下のような効果が生じる。

（１）請求項１記載の発明によれば、演算回路をシリアルに並べることで、同じユニットが利用でき、かつパイプライン化できるので高速化することができる。

【００５２】（２）請求項２記載の発明によれば、前記色変換テーブルは、ｎ色入力の時に２のｎ乗個に分割することにより、補間演算の点数を自由に設定することができる。

【００５３】（３）請求項３記載の発明によれば、前記分割された色変換テーブルが同時に使われない組み合わせについては、パラレルに並べることで、遅延を少なくすることができる。

【００５４】（４）請求項４記載の発明によれば、全体

の色変換テーブルを複数の色変換テーブルに分割して持ち、2のn乗より少ない数で補間する場合、当該色変換テーブルを使用しない部分に0の重みを加えることで、簡単な構成でバイパスするのと同等の効果が得られる。

【0055】（5）請求項5記載の発明によれば、前記補間演算点数を可変することで、入力データに応じて最適な補間演算手段が選択できる。

（6）請求項6記載の発明によれば、演算回路をシリアルに並べることで、同じユニットが利用でき、かつパイプライン化できるので高速化することができる。

【0056】このように、本発明によれば、高速演算が可能な色補間方法及び色補間装置を提供することができる。即ち、第1にメモリの総量を最小限にして、かつ最大の計算能力を実現し、第2に同じ形式のユニットを並べることで、部品及びタイミング設計を容易にし、第3に容易に補間に必要な格子点数を変更することができる。

【図面の簡単な説明】

【図1】本発明の一実施の形態例を示すブロック図である。

【図2】三角錐を用いた4点補間の説明図である。

【図3】LUTの分割の説明図である。

【図4】ビット振り分け器の一実施の形態例を示すブロック図である。

【図5】三角錐コードの説明図である。

【図6】係数発生器の一実施の形態例を示すブロック図である。

【図7】アドレス／重み発生器の一実施の形態例を示すブロック図である。

【図8】差分計算指示器の内容を示す図である。

【図9】差分計算指示器の内容を示す図である。

【図10】副LUTと乗算器の一実施の形態例を示すブロック図である。

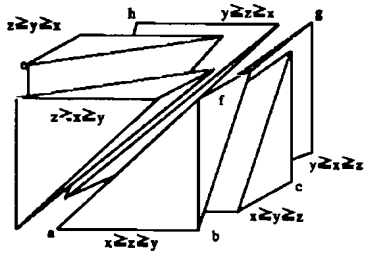
【図11】本発明の他の実施の形態例を示すブロック図である。

【符号の説明】

1　ビット振り分け器
2　係数発生器
3　副LUTと乗算累積器
4　割算器
10、20、30　演算部

【図2】

三角錐を用いた4点補間の説明図

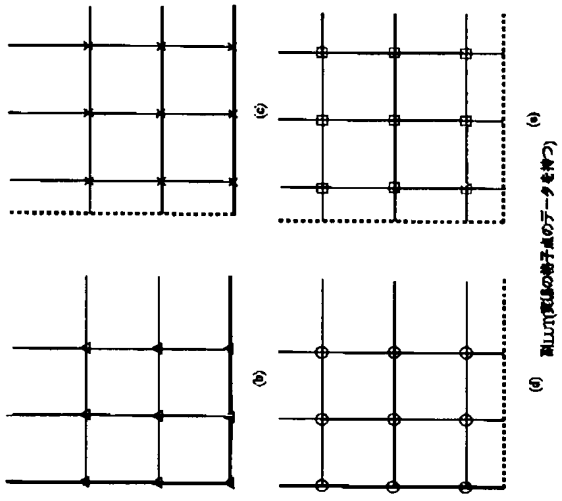


x≧y≧z

Pxyz = Pa (1-x)
+Pb (x-y)
+Pc (y-z)
+Pgz

【図3】

LUTの分割の説明図



【図5】

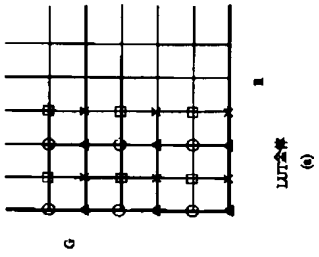
三角錐 コードの説明図

条件	コード
R≧G≧B	0
R≧B≧G	1
G≧R≧B	2
G≧B≧R	3
B≧R≧G	4
B≧G≧R	5

【図9】

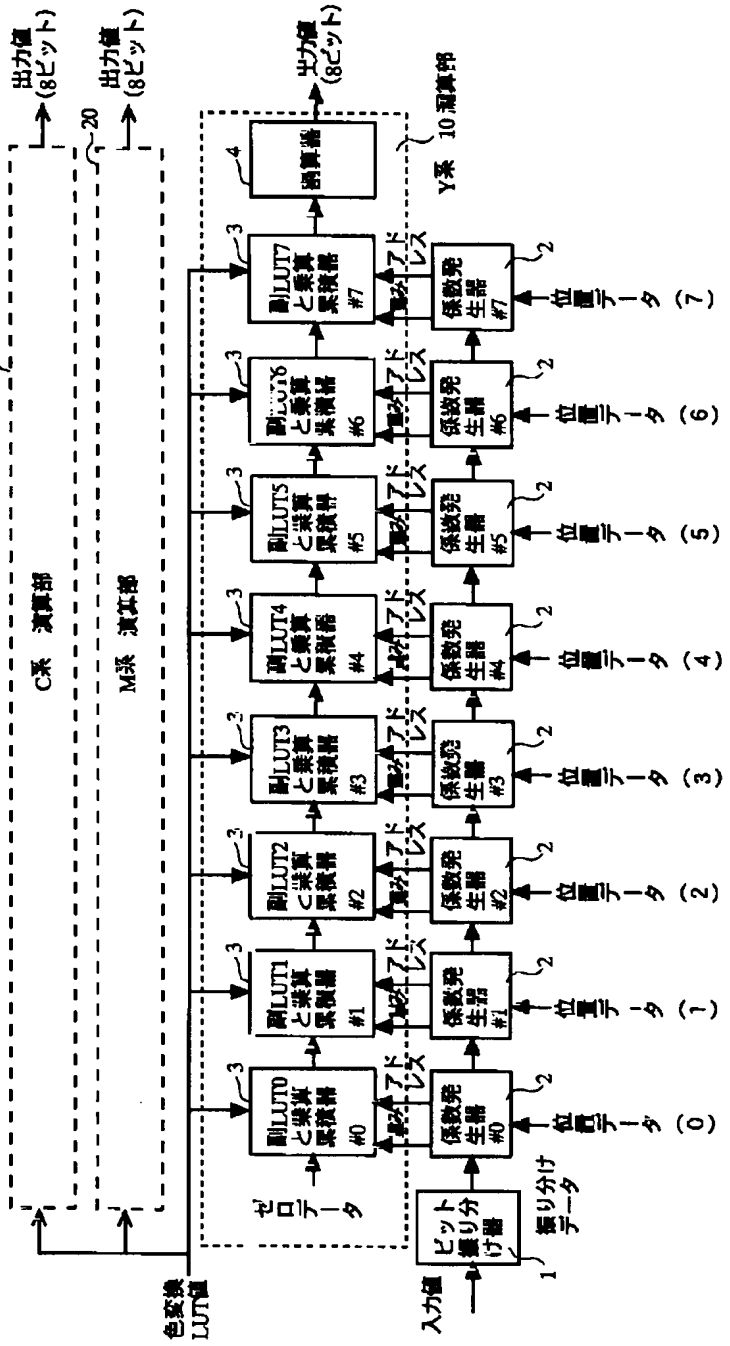
差分計算指示器の内容を示す図

コード	出力
0	0
1	17-R
2	17-G
3	17-B
4	R-G
5	R-B
6	G-R
7	G-B
8	B-R
9	B-G
10	B
11	G
12	R



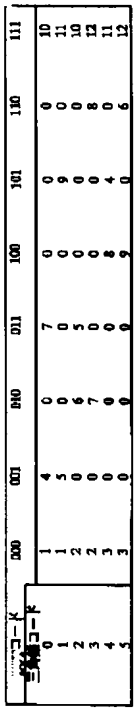
【図1】

本発明の一実施の形態例を示すブロック図



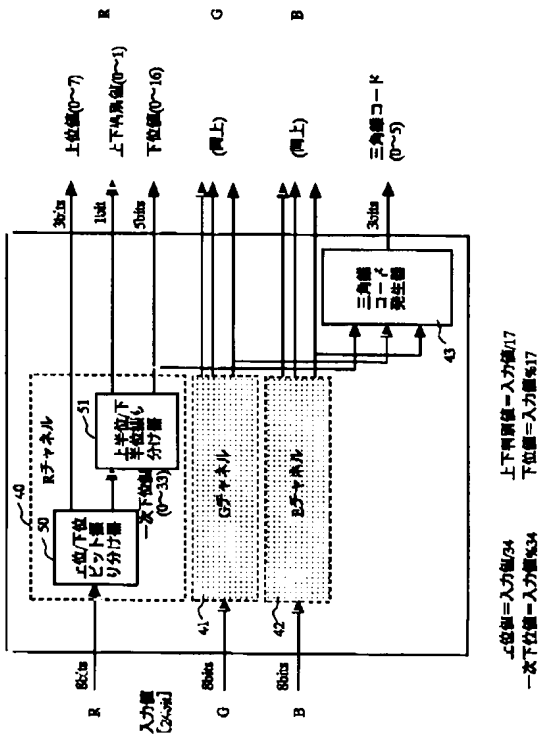
【図8】

差分計算指示器の内容を示す図



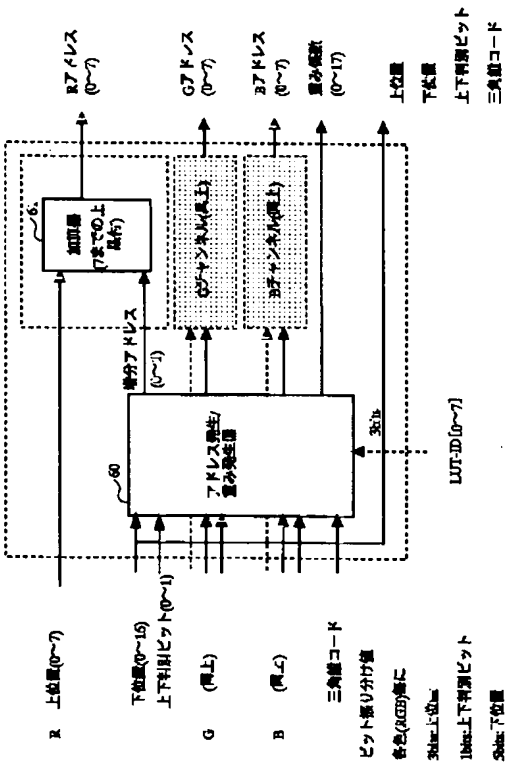
【図4】

ビット振り分け器の一実施の形態例を示すブロック図



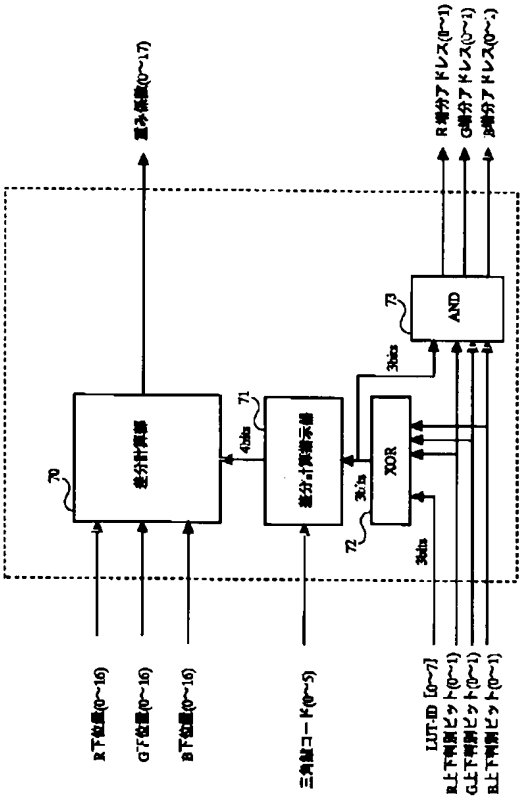
【図6】

係数発生器の一実施の形態例を示すプログラム



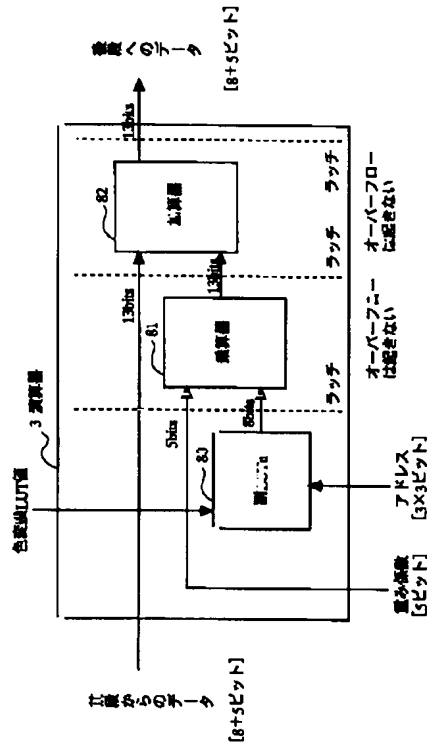
【図7】

アドレス発生／重み発生部の一実施の形態例を示すブロック図



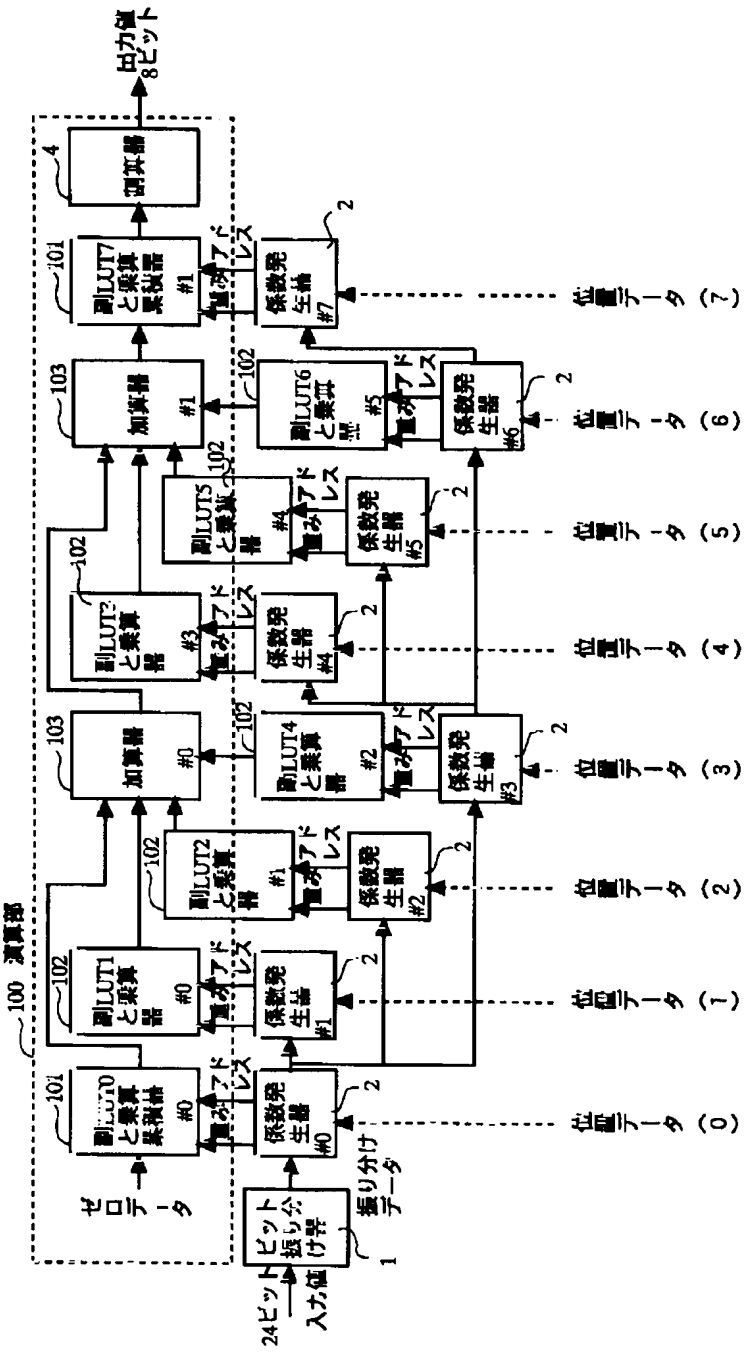
【図10】

乗算LUTと減算器の一実施の形態例を示すブロック図



【図11】

本発明の他の実施の形態例を示すブロック図



フロントページの続き

Fターム(参考) 2C262 AB19 BA01 BC01 BC05 GA25
5B057 AA11 AA20 CA01 CA08 CA12
CA16 CB01 CB08 CB12 CB16
CC01 CE17 CH05 CH07 CH09
5C077 LL18 MP08 NN02 PP32 PP37
PQ13 PQ23 RR06 TT02
5C079 HB01 LB02 MA04 MA12 NA03
NA11 PA03